

5. Map

Map เป็นโครงสร้างข้อมูลในรูปแบบที่เรียกว่า associative array ที่เก็บค่าของสมาชิก คือ ค่า key และ value โดยค่า key ต้องเป็นค่าที่ไม่ซ้ำกัน ในไลบรารี STL โครงสร้างข้อมูล map เก็บค่าแบบเรียงลำดับ ส่วน unordered_map เก็บค่าแบบไม่เรียงลำดับ

ฟังก์ชันของโครงสร้างข้อมูล map ใน STL

Operation	Description
<code>map<keytype, valuetype> m</code>	Creates an empty map without any elements
<code>m.insert()</code>	Inserts a new element with key/value pairs
<code>m.find(key)</code>	Return an iterator if key is found. If no such key is found, past-the-end iterator is returned. (see example in cpp reference)
<code>m.erase(key)</code>	Removes the elements using key
<code>m.erase(iterator)</code>	Removes the elements using iterator position
<code>m.lower_bound(k)</code>	Returns an iterator pointing to the first element in the container whose key is not considered to go before <i>k</i>
<code>m.upper_bound(k)</code>	Returns an iterator pointing to the first element in the container whose key is considered to go after <i>k</i> .

การประกาศการใช้ map

```
map<keytype, valuetype> variable_name
```

เช่น

```
map<char, int> mymap;  
map<string, int> mymap;  
map<int, int> mymap;
```

ตัวอย่างโปรแกรมที่ 5.1 การเพิ่มสมาชิกใน map

```
1  #include <stdio.h>
2  #include <map>
3  using namespace std;
4  int main (){
5
6      map<char,int> mymap;
7
8      mymap['a']=100;
9      mymap['b']=200;
10     mymap['c']=300;
11     mymap['d']=400;
12
13     map<char, int>::iterator it;
14     for(it=mymap.begin(); it!=mymap.end(); ++it){
15         printf("%c => %d\n", it->first, it->second);
16     }
17     return 0;
18 }
```

ผลลัพธ์

```
a => 100
b => 200
c => 300
d => 400
```

ตัวอย่างโปรแกรมที่ 5.2 การเพิ่มสมาชิกใน map

```
1.  #include <iostream>
2.  #include <map>
3.  #include <string>
4.  using namespace std;
5.  int main (){
6.      map<string,int> mymap;
7.      // insert a new entry into the map
8.      mymap.insert(pair<string,int>("Tim",30));
9.      mymap.insert(make_pair("Noah",38));
10.     mymap.insert({"Sunny",34});
11.     mymap["Dennis"] = 39;
12.
13.     map<string, int>::iterator it;
14.     for(it=mymap.begin(); it!=mymap.end(); ++it){
15.         cout << it->first << " => " << it->second << endl;
16.     }
17.     return 0;
18. }
```

ผลลัพธ์

```
Dennis => 39
Noah => 38
Sunny => 34
Tim => 30
```

ตัวอย่างโปรแกรมที่ 5.3 การเปลี่ยนแปลงค่าใน map

```
1  #include <iostream>
2  #include <map>
3  #include <string>
4
5  using namespace std;
6
7  int main (){
8      map<string,int> mymap;
9
10     mymap.insert({"Tim",30});
11     mymap.insert({"Dennis",35});
12     mymap.insert({"Noah",37});
13
14     //map<string, int>::iterator it;
15     for(auto it=mymap.begin(); it!=mymap.end(); ++it){
16         cout << it->first << " => " << it->second << endl;
17     }
18
19     mymap["Tim"] = 40;
20     mymap["Dennis"] = 45;
21     mymap["Noah"] = 47;
22
23     cout << "*** After modifying values ***\n" ;
24     for(auto it=mymap.begin(); it!=mymap.end(); ++it){
25         cout << it->first << " => " << it->second << endl;
26     }
27     return 0;
28 }
```

ผลลัพธ์

```
Dennis => 35
Noah => 37
Tim => 30
*** After modifying values ***
Dennis => 45
Noah => 47
Tim => 40
```

ตัวอย่างโปรแกรมที่ 5.4 การลบสมาชิกใน map

```
1  #include <iostream>
2  #include <map>
3  #include <string>
4
5  using namespace std;
6
7  int main () {
8      map<string,int> mymap;
9
10     mymap.insert({"Tim",30});
11     mymap.insert({"Dennis",35});
12     mymap.insert({"Noah",37});
13
14     for(auto it=mymap.begin(); it!=mymap.end(); ++it){
15         cout << it->first << " => " << it->second << endl;
16     }
17
18     mymap.erase("Tim");
19     cout << "*** After deleting \"Tim\" ***\n" ;
20     for(auto it=mymap.begin(); it!=mymap.end(); ++it){
21         cout << it->first << " => " << it->second << endl;
22     }
23     return 0;
24 }
```

ผลลัพธ์

```
Dennis => 35
Noah => 37
Tim => 30
*** After deleting "Tim" ***
Dennis => 35
Noah => 37
```

ตัวอย่างโปรแกรมที่ 5.5 การลบสมาชิกใน map ด้วยการใช้ iterator

```
1  #include <iostream>
2  #include <map>
3  using namespace std;
4  int main()
5  {
6      map<int, int> mymap;
7
8      mymap.insert({3,40});
9      mymap.insert({1,50});
10     mymap.insert({4,70});
11     mymap.insert({2,30});
12     mymap.insert({7,80});
13     mymap.insert({6,40});
14     mymap.insert({5,50});
15     for(auto it=mymap.begin(); it!=mymap.end(); it++){
16         cout << it->first << " => " << it->second << endl;
17     }
18     // remove all elements up to
19     // element with key=4
20     mymap.erase(mymap.begin(), mymap.find(4));
21     cout << "\nmymap: after removing elements less than key=4\n";
22     for(auto it=mymap.begin(); it!=mymap.end(); it++){
23         cout << it->first << " => " << it->second << endl;
24     }
25     return 0;
26 }
```

ผลลัพธ์

```
1 => 50
2 => 30
3 => 40
4 => 70
5 => 50
6 => 40
7 => 80
```

```
mymap: after removing elements less than key=4
4 => 70
5 => 50
6 => 40
7 => 80
```

ตัวอย่างโปรแกรมที่ 5.6 การทำสำเนา map

```
1  #include <iostream>
2  #include <map>
3  using namespace std;
4  int main()
5  {
6      map<int, int> mymap;
7      mymap.insert({3,40});
8      mymap.insert({1,50});
9      mymap.insert({4,70});
10     mymap.insert({2,30});
11     mymap.insert({7,80});
12     mymap.insert({6,40});
13     mymap.insert({5,50});
14     cout << "*** printing mymap ***\n" ;
15     for(auto it=mymap.begin(); it!=mymap.end(); it++){
16         cout << it->first << " => " << it->second << endl;
17     }
18     // assigning the elements from mymap to mymap2
19     map<int, int> mymap2(mymap.begin(), mymap.end());
20     cout << "*** printing mymap2 ***\n" ;
21     for(auto it=mymap2.begin(); it!=mymap2.end(); it++){
22         cout << it->first << " => " << it->second << endl;
23     }
24     return 0;
25 }
```

ผลลัพธ์

```
*** printing mymap ***
1 => 50
2 => 30
3 => 40
4 => 70
5 => 50
6 => 40
7 => 80
*** printing mymap2 ***
1 => 50
2 => 30
3 => 40
4 => 70
5 => 50
6 => 40
7 => 80
```

ตัวอย่างโปรแกรมที่ 5.7 การใช้ lower_bound() และ upper_bound()

```
1  #include <iostream>
2  #include <map>
3  using namespace std;
4  int main()
5  {
6      map<int, int> mymap;
7      mymap.insert({3,40});
8      mymap.insert({1,50});
9      mymap.insert({4,70});
10     mymap.insert({2,30});
11     mymap.insert({7,80});
12     mymap.insert({6,40});
13     mymap.insert({5,50});
14
15     auto lo5 = mymap.lower_bound(5);
16     cout << "mymap.lower_bound(5) : ";
17     cout << lo5->first << " => " << lo5->second << "\n";
18     auto up5 = mymap.upper_bound(5);
19     cout << "mymap.upper_bound(5) : ";
20     cout << up5->first << " => " << up5->second << "\n";
21     return 0;
22 }
```

ผลลัพธ์

```
mymap.lower_bound(5) : 5 => 50
mymap.upper_bound(5) : 6 => 40
```

6. Unordered map

เปรียบเทียบระหว่าง map และ unordered_map

	Map	Unordered map
Ordering	เก็บค่าเรียงลำดับตาม key	ไม่เรียงลำดับ
Implementation	Binary search tree	Hash table
Search time	$\log(n)$	$O(1)$ -> Average $O(n)$ -> Worst Case
Insertion time	$\log(n)$ + rebalance	เช่นเดียวกับ search
Deletion time	$\log(n)$ + rebalance	เช่นเดียวกับ search

ตัวอย่างที่ 6.1 การเพิ่มสมาชิกใน map

```
1.  #include <stdio.h>
2.  #include <unordered_map>
3.  using namespace std;
4.  int main () {
5.      unordered_map<char,int> umap;
6.
7.      umap['d']=100;
8.      umap['c']=200;
9.      umap['a']=300;
10.     umap['b']=400;
11.
12.     unordered_map<char, int>::iterator it;
13.     for(it=umap.begin(); it!=umap.end(); it++){
14.         printf("%c => %d\n", it->first, it->second);
15.     }
16.     return 0;
17. }
```

ผลลัพธ์

```
b => 400
a => 300
c => 200
d => 100
```


ตัวอย่างที่ 6.2 การเพิ่มสมาชิกใน map ด้วยวิธีต่างๆ และ key มีชนิดเป็น string

```
1.  #include <iostream>
2.  #include <unordered_map>
3.  #include <string>
4.  using namespace std;
5.  int main () {
6.      unordered_map<string, int> umap;
7.      umap.insert(pair<string, int>("Tim", 30));
8.      umap.insert(make_pair("Noah", 38));
9.      umap.insert({"Sunny", 34});
10.     umap["Dennis"] = 39;
11.     for(auto it=umap.begin(); it!=umap.end(); ++it){
12.         cout << it->first << " => " << it->second << endl;
13.     }
14.     return 0;
15. }
```

ผลลัพธ์

```
Dennis => 39
Sunny => 34
Noah => 38
Tim => 30
```

ตัวอย่างที่ 6.3 การเปลี่ยนแปลงค่าใน map

```
1.  #include <iostream>
2.  #include <unordered_map>
3.  #include <string>
4.  using namespace std;
5.  int main()
6.  {
7.      map<unsigned int, string> employer;
8.      unordered_map<unsigned int, unsigned> salary;
9.      employer[1001] = "Dennis";
10.     employer[1002] = "Tim";
11.     employer[1003] = "Noah";
12.     employer[1004] = "Sunny";
13.     salary[1001] = 24320;
14.     salary[1002] = 35520;
15.     salary[1003] = 74350;
16.     salary[1004] = 44510;
17.     for(auto e: employer) {
18.         cout << "Name: " << e.second << "\nID: " << e.first <<
19.         "\n";
20.         cout << "Salary: " << salary[e.first] << "\n";
22.     }
23.     unsigned total_salary = 0;
24.     for(auto s: salary)
25.         total_salary += s.second;
26.     cout << "Total " << total_salary << endl;
27.
28.     return 0;
29. }
```

ผลลัพธ์

```
Name: Dennis
ID: 1001
Salary: 24320
Name: Tim
ID: 1002
Salary: 35520
Name: Noah
ID: 1003
Salary: 74350
Name: Sunny
ID: 1004
Salary: 44510
Total 178700
```